Analysis of Discrete-Time Systems

Notes on MATLAB and the Four Systems

- **MATLAB data:** are always in the form of a matrix. In this project, a discrete-time signal which starts at time $n = 0$ and ends at time $n = N - 1$ (total length $= N$) is represented by a *row* vector (i.e., a $1 \times N$ matrix). For example, the signal $x[n] = 5\delta[n] + 3\delta[n-2] - \delta[n-4]$ (total length $N = 5$) is represented as:

```
>> x = [5 0 3 0 -1];
```

  Thus, the first number is assumed to be $x[0]$, the second number is assumed to be $x[1]$, and so on. The last number is assumed to be $x[N-1]$.

- **dtplot:** is a discrete-time plotting program written by N. Phamdo. It is based on the *stem* plot of MATLAB. Try plotting the signal $x[n] = 5\delta[n] + 3\delta[n-2] - \delta[n-4]$ by doing the following:

```
>> x = [5 0 3 0 -1];
>> dtplot(x)
```

  The input to dtplot must always be a row vector. The program plots the signal $x[n]$ from $n = -3$ to $n = N + 2$ (the first and last 3 values are set to zero) and scales the axes accordingly.

- **The Four Systems:** are implemented in the functions: *system1, system2, system3, system4.* For example, if you want to plot the output of each system corresponding to the input signal, $x[n] = 5\delta[n] + 3\delta[n-2] - \delta[n-4]$, do

```
>> x = [5 0 3 0 -1];
>> y1 = system1(x);
>> y2 = system2(x);
>> y3 = system3(x);
>> y4 = system4(x);
>> dtplot(y1)
>> dtplot(y2)
>> dtplot(y3)
>> dtplot(y4)
```

  Note that the input to each system must be a row vector. The length of the output signal may be different than the length of the input signal. To check the output length, do (without the semicolon):

```
>> size(y1)
```

- **The Following Examples:** illustrate how you generate long test input signals. In all cases, we assume that the length of the signal is $N = 100$.

  (a) $x[n] = \delta[n]$ (viewed as length 100):

```
>> N=100;
>> x=zeros(1,N);
>> x(1)=1;
>> dtplot(x);
```

  (b) $x[n] = u[n] - u[n-N]$

```
>> x=ones(1,N);
>> dtplot(x);
```

(c) $x[n] = n(u[n] - u[n-N])$

```
>> x=0:N-1;
>> dtplot(x);
```

(d) $x[n] = \cos(\pi n/20)(u[n] - u[n-N])$

```
>> n=0:N-1;
>> x=cos(pi/20*n);
>> dtplot(x);
```

(e) $x[n] = (0.9)^n(u[n] - u[n-N])$

```
>> n=0:N-1;
>> x=power(0.9,n);
>> dtplot(x);
```

(f) $x[n]$ is a random signal with values between 0 and 1

```
>> x=rand(1,N);
>> dtplot(x);
```

- **Built-In Help:** is available in MATLAB. Try

```
>> help
>> help general
>> help elfun
```